Journal of mathematics and computer science 14 (2015), 193-204

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

# Comparing Imperialist Competitive Algorithm with Backpropagation Algorithms for Training Feedforward Neural Network

**Maryam Zanganeh[1,*], Seyed Javad Mirabedini[1,+]**

[1]Department of computer engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

[*]*m.zangeneh@ymail.com*

[+]*j.mirabedini@gmail.com*

**Article history:**
**Received November 2014**
**Accepted December 2014**
**Available online December 2014**

***Abstract***

Artificial Neural Networks (ANN) and evolutionary algorithms are two relatively young research areas that were subject to a steadily growing interest during the past years. The use of ANN has been proved to be a cost-effective technique. It is very important to choose a suitable algorithm for training a neural network. Mostly Back Propagation (BP) algorithm is a gradient descent algorithm (a first-order optimization algorithm) on the error space, which most likely gets trapped into a local minimum and has very slow convergence. This shortcoming can be removed by global searching ability of the evolutionary algorithms such as Imperialist Competitive Algorithm (ICA) which is a new evolutionary algorithm based on the human's socio-political evolution. This investigation provides a comparison between training a neural network with BP algorithms used for training Feed-forward Neural Networks (FNN) and ICA. Among the BP algorithms, Gradient descent, Levenberg–Marquardt, Conjugate gradient descent, Resilient, BFGS Quasi-newton, and One-step secant algorithm are tested then the obtained results will be compared with the results of training the neural network with ICA. Also, Accuracy and Mean Squared Error (MSE) are the main measures selected to assess both models. Also the MSE was used as a criterion to specify optimum number of neurons in the hidden layer. The results showed that ICA approach outperforms the BP for training neural network models.

**Keywords:** Imperialist Competitive Algorithm, Backpropagation, Artificial Neural Network.

## 1. Introduction

ANN is an information processing model inspired by the biological nervous systems, such as brain, which processes information [1]. The novel structure of neurons (the information processing elements) working together to solve particular problems, is the key element of this model. Most of the present works in ANN concern the development, characterization and extension of mathematical neural

network models. Generally neural network learning is a nonlinear minimization issue with many local minimum [2] and depends on network weights, learning rules and architecture [3].

The FNN with the BP training algorithm is the most common neural network architecture. FNN is a network structure in which the information or signals will scatter only in one direction, from input to output. Any nonlinear continuous function can be approximated to an arbitrary accuracy by a three layered FNN with BP [4]. Among different FNNs, Multilayer Perceptron (MLP) which is widely used for pattern classification, recognition, prediction and approximation was chosen in this study. By optimizing the weights for each node interconnection and bias terms, the network is trained; until the value of output neurons are close to the real outputs. This paper shows the performance comparison of the MLP networks using various BP learning algorithms with a new evolutionary algorithm called ICA. The BP learning algorithms used are Gradient descent, Levenberg-Marquardt (LM), Conjugate gradient descent, Resilient, BFGS Quasi-newton, and One-step secant.

Following this introductory section, the rest of the paper is organized as follows: In section 2 related works are presented then in section 3 the ANN is defined while the related subsection is allocated to explaining of two different algorithms for training ANNs; Gradient decent and ICA. German dataset used in this paper is stated in section 4. Model performance assessment criteria, Accuracy and MSE, are specified in section 5, then in section 6 our proposed methodology is analyzed clearly. This paper is finished by illustrating the results in section 7 and the last section is allocated to the conclusions of this study.

## 2. Related Works

There are many researches in the field of training neural networks; Iftikhar et al. in 2008 used three ANNs with different algorithms to the problem of intrusion detection in computer and network systems. The learning algorithms considered were the standard, the resilient and the batch BP algorithm. They concluded that the resilient algorithm had a better performance than the application [5]. Noori et al. in 2010 to predict monthly stream flow evaluated different training functions on ANN operation. The result showed that scaled conjugate gradient models had the best performance in wet and arid periods [6]. Mirjalili et al. in 2011 proposed learning method and compared it with PSO and GA-based learning algorithms. The results proved the high performance of this new learning algorithm for large numbers of training samples [7]. Ahmadi et al. in 2013 presented a new method based on Fuzzy logic, ANN and ICA for oil rate prediction of wells. The results showed the compatibility, robustness and effectiveness of the ICA-ANN model [8]. Duan and Huang in 2014 presented a novel hybrid method for the globally optimal path planning which was based on an ANN trained by ICA. The comparative results with Artificial Bee Colony (ABC) algorithm showed that their proposed approach could not only reduce the uncertainty of the evolutionary computation caused by the probability model, but also avoided falling into local point with much quicker speed [9]. Hoseini and Al Khaled in 2014 reviewed the underlying ideas of how ICA emerged and its application to the engineering disciplines mainly on industrial engineering [10].

## 3. Artificial Neural Networks

ANNs have several neurons which are connected to each other. Each connection between the neurons has a particular weight, which shows how much the output of the neuron will influence the input to the next neuron. Also, each neuron has a weight of its own known as a bias term which determines the effect of the neuron itself. The number of nodes in input layer of FNNs presents the number of inputs in the process, whereas the number of output nodes shows the number of real output [11-14].
Among the different FNNs, we chose MLP neural network, which is widely used for pattern classification, recognition, prediction and approximation. In fact, MLP can solve problems which are

not linearly separable. A typical MLP network consists of three sequential and fully connected layers such as input, hidden, and output layer. The input layer accepts characteristic parameters. The role of a hidden layer is to process and transmit the information from input layer toward the output layer concerning its connections with the input layer and the weights of those connections. The final layer that interacts with the output to present the processed data is known as the output layer [15]. The performance of an MLP model is highly dependent on the architecture of the network because if the number of neurons in the hidden layer is excessive or insufficient, it most likely leads to the problems of bad generalization and over-fitting. Literature review shows that there is no analytical method to determine the optimum number of neurons in the hidden layer. During the last decade, some meta-heuristic techniques have been introduced to find it, but most of researchers used trial and error approach [16]. The information in an ANN is stored in its weights; hence training an ANN requires the weights to be determined. There are several existing algorithms that deal with this process. BP is a common method used for FNNs. It computes the error on all the training pairs and in order to fit the desired output, it adjusts the weights. This process is done in several periods until the error stops to decrease or when the total error on the training set become small enough. In fact, it puts supervised learning to use in which the network is trained using data for which inputs as well as target outputs are observed.

### 3.1. Gradient descent

The ANN is an adaptive nonlinear system competent to estimate any function. Theoretically, a continuous function can be estimated to a desired accuracy on any compact set by a neural network [17-19]. As mentioned before, BP algorithms are common methods used for FNNs. Among them, the most effective method with respect to the training accuracy is the LM method which is well known and popularly demonstrated in the neural networks literature. The LM algorithm is a repetitive technique that identifies the minimum of a multivariate function that is described as the sum of squares of non-linear real-valued functions [20, 21]. A precise description of the LM neural network training algorithm has been presented by Hagan and Menhaj [22]. Also, because of its proper convergence capabilities, the LM method is usually preferred over many other optimization methods, in many optimization applications.
BP is based on the gradient descent algorithm which is a very popular optimization method, but it suffers from slow convergence and susceptibility to a local minimum. This deficiency can be eliminated by an exploration ability of the evolutionary algorithms such as ICA with two great aspects; high power of this algorithm to search the global optimization also when facing with nonlinear optimization problems and fast convergence speed.

### 3.2. Imperialist Competitive Algorithm

ICA as an evolutionary optimization method is inspired by imperialistic competition and is based on the behavior of imperialists in their attempt to overcome colonies. Similar to other evolutionary algorithms, ICA begins with an initial population and is separated into two types: the colonies and the imperialists (the ones with the best objective function values). All of these colonies, based on their power, are shared among these imperialists. The power of each country is proportionate to its cost in reverse, i.e. the more powerful an imperialist is, the more colonies it will dominate [23]. In the language of ICA, an empire is formed with an imperialist with its colonies. Therefore, imperialistic competition among these empires builds the proposed evolutionary algorithm [24]. When the colonies begin to change their culture in such a way that it becomes more similar to the one of their governing imperialist over time, this is one of the characteristics of the interaction between imperialist powers and their colonies. By moving the colonies towards their imperialist, this process is implemented in

ICA called assimilation policy which the imperialistic countries have reached after the 19th century. During this event, it is possible that a colony becomes more authoritative than it's imperialist, in this case, by taking the place of the imperialist, the imperialist will become one of its colonies. Furthermore, during imperialistic competition, the most authoritative empires tend to enhance their power, whereas the weaker ones tend to fall in. These two mechanisms result the algorithm to progressively converge into a condition in which there exists only one empire and colonies have the same objective function. The flowchart of the ICA is shown in Figure 1.

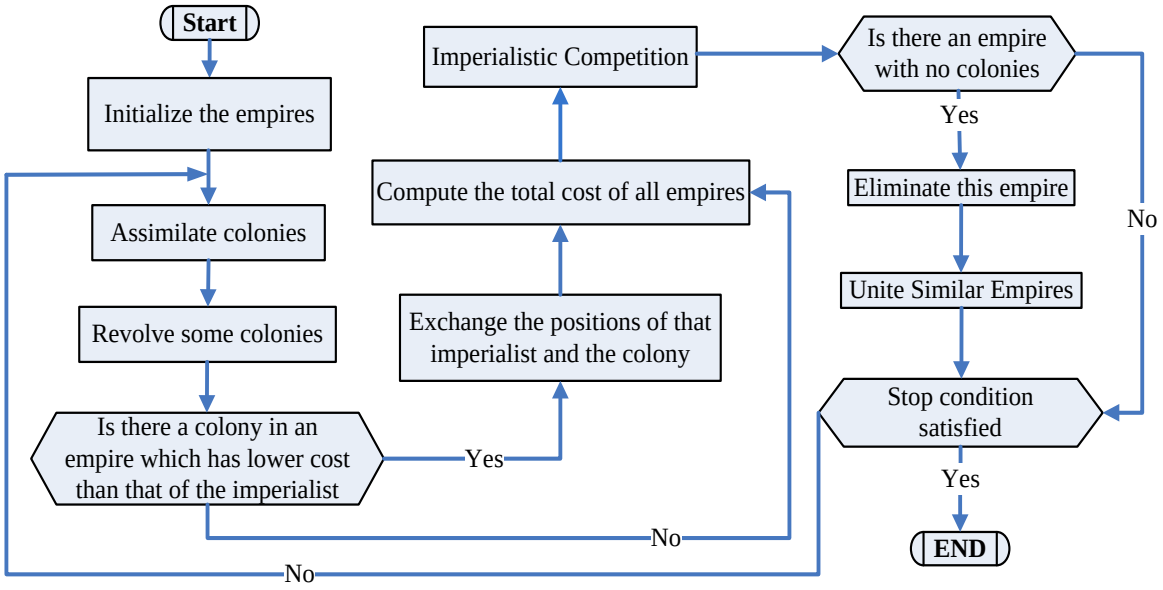


**Figure 1. Flowchart of the Imperialist Competitive Algorithm**

Based on assimilation policy, after sharing all colonies among imperialists and making the initial empires, these colonies begin moving into their related imperialist territory which is shown in Figure 2. By this motion, as said in Equation 1, x is a random variable with uniform distribution between 0 and β ×d. To examine different points around the imperialist, a random amount of deviation added to the direction of the movement shown in Equation 2 is named θ, which is a random number with uniform distribution between -γ and +γ and adjusts the derivation from the original direction.

$$x \sim \cup (0, \beta \times d) \quad (1)$$

$$\theta \sim \cup (-\gamma, \gamma) \quad (2)$$

Where d is the distance between the imperialist and its colony and $\beta$ is a positive number between (1,2). A value about 2 for $\beta$ and about π/4 (Rad) for $\gamma$ results in good convergence of countries to the global minimum.

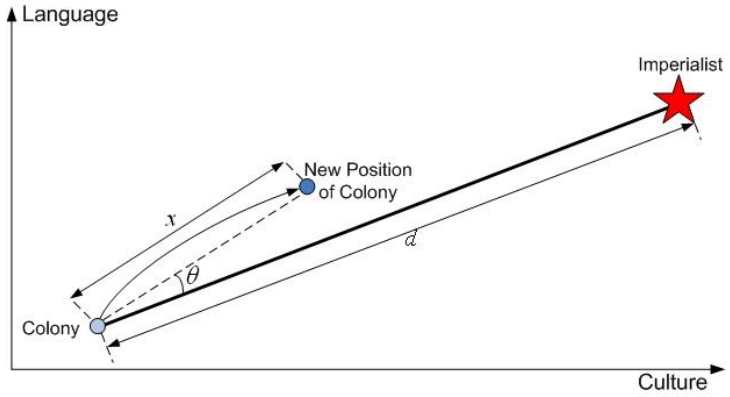


**Figure 2. Movement of colonies toward their relevant imperialist in a randomly deviated direction**

In the terminology of ICA, revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis. Figure 3 shows the revolution in Culture-Language axis. The revolution increases the exploration of the algorithm and prevents the early convergence of countries to local minimums. The revolution rate in the algorithm indicates the percentage of colonies in each colony which will randomly change their position. A very high value of revolution decreases the exploitation power of algorithm and can reduce its convergence rate.

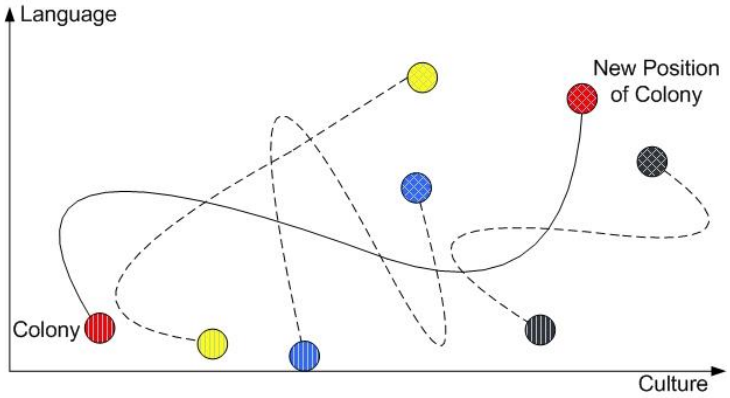


**Figure 3. Revolution; a sudden change in socio-political characteristics of a country**

While moving toward the imperialist, a colony might reach to a position with lower cost than the imperialist. In this case, the position of the colony and imperialist will exchange. Then the algorithm will continue by the imperialist in the new position and the colonies will be assimilated by the imperialist in its new position. Figure 4 depicts the position exchange between a colony and the imperialist.

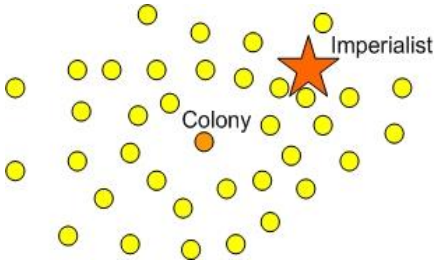


**Figure 4(a). Exchanging the positions of a colony and the imperialist**

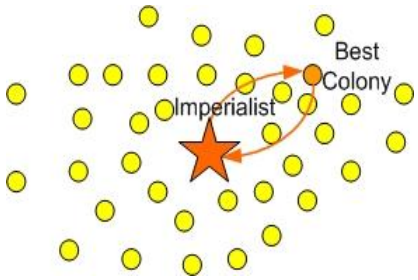


**Figure 4(b). The entire empire after position exchange**

The power of these countries is appeared by calculating the cost of objective function for each empire. Equation 3 shows how the power of empires can be calculated:

$$T.C._n = Cost(imperialist_n) + \xi\ mean\{Cost(colonies\ of\ empire_n)\} \quad (3)$$

Where the total cost of the $n$th empire is $T.C._n$ and $\xi$ is a positive number which is considered to be less than 1. A little value for $\xi$ causes the total power of the empire to be determined by just the imperialist and increasing it will increase the role of the colonies in determining the total power of an empire.

Next step is imperialistic competition, wherein all empires try to take control over colonies of the other empires. So, the most authoritative empire will take possession of a colony from the weakest empire. Figure 5 depicts how imperialistic competition can be modeled.

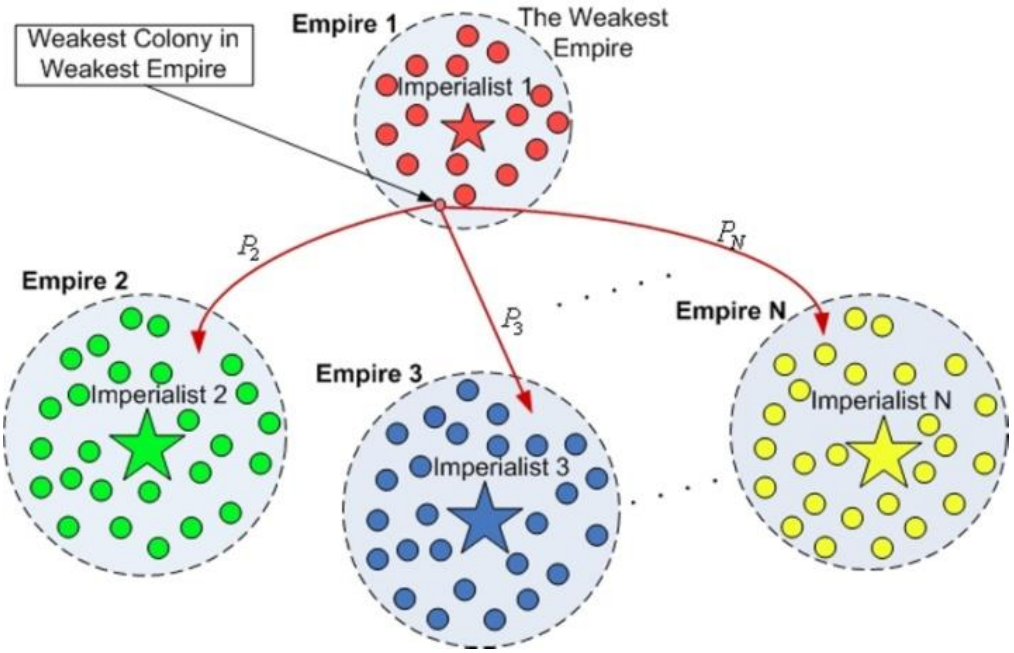


**Figure 5. Imperialistic competition. The most powerful empire will possess the weakest colony of the weakest empire.**

After a few repetitions, by collapsing the powerless empires, their colonies will be shared among the rest of the empires in the imperialist competition. In this new world, there will be no difference, neither among colonies, nor between the imperialist and colonies. It means that the position and cost of the imperialist are such as the position and cost of colonies which are controlled by the same imperialist. In this case, the imperialistic competition ends and the algorithm finishes. Position and cost of remained imperialist respectively show the optimized variables and the result of the problem [25].

## 4. Dataset

In this study, we use the UCI machine learning database, one of the 100 reliable databases, which has the most references in scientific papers available via http://archive.ics.uci.edu/ml/datasets.html. The German dataset is the credit dataset used in this study to evaluate the models. Table1 shows a summary of the main features of this credit dataset.

**Table 1. Some Details of dataset used in study**

| No. | Dataset | No. of Attributes | No. of Good Instances | No. of Bad Instances |
|---|---|---|---|---|
| 1 | German | 21 | 700 | 300 |

In data normalization process the input data are normalized between 0 and 1. This is performed by finding the maximum number in each column (feature) for all instances and dividing the rest of the entries in each column to its maximum value.

## 5. Model performance assessment criteria

The performance assessment criteria used in this study include Accuracy and MSE. Accuracy indicates the proportion of the correctly classified cases on a particular dataset and is defined by Equation (4);

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{4}$$

Where, TP (True Positive): is the proportion of positive cases that are correctly diagnosed as positive; FP (False Positive): is the proportion of negative cases that are wrongly diagnosed as positive; FN (False Negative): is the proportion of positive cases that are wrongly diagnosed as negative; TN (True Negative): is the proportion of negative cases that are correctly diagnosed as negative.

Another performance assessment criteria is the difference between the actual and predicted value by the model, named MSE. This measure is computed by Equation (5) and our target is to minimize this value;

$$\text{e} = \hat{\text{y}} - \text{y} \Rightarrow \text{MSE} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{e}_\text{i}^2 \tag{5}$$

Where, $e$ is the difference between $\hat{y}$ as predicted value and $y$ as the actual value.

## 6. Methodology

In this study, at first, in order to find the best BP neural network, different learning algorithms were used which are shown in Table 2. ANNs are constructed based on learning algorithm, activation function and the number of neurons in their hidden layer as the parameters. Log-sigmoid and

hyperbolic tangent sigmoid were used as activation functions. To assess the reliability and validity of the model, the BP learning algorithms are examined by certain numbers of training, validation and testing data. In this study, 60% of data considered as training data for building the model, 10% as validation data and 30% were considered for testing the model. Learning rate of 0.01 and a momentum coefficient of 0.9 was considered in this work for BP Learning.

**Table 2. Different learning algorithms**

| No. | Abbreviation | Learning Algorithm Description |
|---|---|---|
| 1 | LM | Levenberg-Marquardt |
| 2 | GD | Gradient Descent |
| 3 | CGD | Conjugate Gradient Descent |
| 4 | R | Resilient |
| 5 | BFGS | BFGS Quasi-newton |
| 6 | OS | One-step Secant |

Another factor which should be considered in constructing an ANN is the number of neurons present in the hidden layer. Each of these six networks with considering the learning ratio and activation function runs and trains for different numbers of neurons from 20 up to 50 within their hidden layer. Training process performs 10 times for each number of these neurons as well. The optimum number of neurons in the hidden layer is specified according to the minimum value of MSE on test data. We compare the performance of the neural network model and then select the ideal neural model.

Secondly, to enhance the convergence rate and learning process, an Imperialist Competitive Algorithm Artificial Neural Network (ICA-ANN) method was used to train the neural network by using the dataset of 1000 credit applicants with 21 attributes. The learning process includes finding a set of weights to minimize the learning error. A set of weights for current iteration shows the position of each country and the number of weights associated with the network represents the coordinate of each country. The performance of ICA-ANN is for getting the best set of weights (country positions). By considering the sum of weights and biases as $d$, however each country presents a possible answer in the $d - dimensional$ space of the problem, the size of the search space is the total number of weights and biases. The goal in this algorithm is to minimize a learning error (cost function) which is calculated by using MSE. In this study, the simulation performance of the ICA-ANN model was evaluated on the basis of MSE and Accuracy. Generally, lowest value of MSE and highest value of accuracy is desirable for selecting best model.

The iterative approach of ICA can be described by the following steps:

1. Select some random points on the function and initialize the empires.
2. Move the colonies toward their relevant imperialist (Assimilation).
3. Randomly change the position of some colonies (Revolution).
4. If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
5. Unite the similar empires.
6. Compute the total cost of all empires.
7. Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (imperialistic competition).
8. Eliminate the powerless empires.
9. If stop conditions are satisfied, stop, if not go to 2.

Parameters used for running ICA in this study are shown in Table 3:

**Table 3. Parameters used for running ICA**

| Parameter | Value |
|---|---|
| No. of empires | 8 |
| No. of countries | 80 |
| Assimilation coefficient ($\beta$) | 2 |
| colonies mean cost coefficient ($\xi$) | 0.05 |
| Revolution rate | 0.2 |
| No. of decades | 200 |
| Assimilation Angle Coefficient ($\gamma$) | 0.5 |
| Damp Ratio | 0.99 |

## 7. Results

At first, the performance of the ANN Back-propagation (ANN-BP) models was evaluated based on the optimum number of neurons in the hidden layer calculated according to Accuracy criterion, i.e. the models have the highest Accuracy for this number of neuron. The obtained results of implementation of the MLP neural network models with six BP learning algorithms are listed in detail in Table 4, which shows that the preferred ANN is the one using LM as learning algorithm. In order to simulate the considered MLP neural network, MATLAB version R2012a software was used.

**Table 4. Implementation results of different BP learning algorithms**

| No. | Learning Algorithm | Optimum No. of Neurons in Hidden Layer | Accuracy |
|---|---|---|---|
| **1** | LM | 22 | 77.10% |
| **2** | GD | 48 | 72.50% |
| **3** | CGD | 33 | 75.90% |
| **4** | R | 24 | 75.70% |
| **5** | BFGS | 47 | 75.30% |
| **6** | OS | 20 | 76% |

As can be observed, among these six models, LM significantly is the best and GD is the worst model. Hence, the performance of LM Back-propagation (LMBP) learning algorithm is compared with ICA. The MSE changes to the number of neurons within the hidden layer with LMBP is shown in figure 6 for German dataset when minimum MSE has been occurred.

In the following the performance of the proposed ICA for training neural network (ICA-ANN) is evaluated. The obtained results of implementation of the ICA-ANN model with mentioned learning ratio and activation functions for optimum number of neurons in the hidden layer are listed in detail in Table 5, while these results are comparing with the LMBP-ANN model.

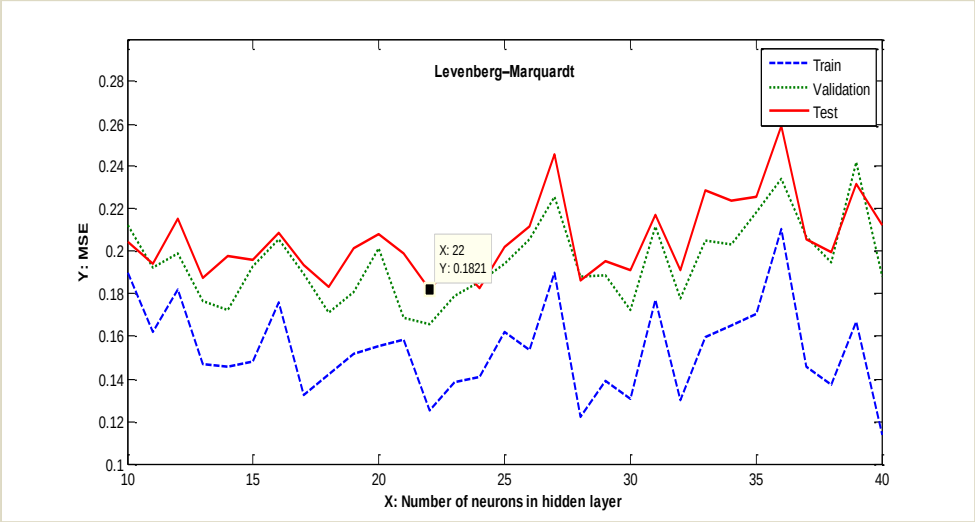


**Figure 6. Optimum number of neurons in the hidden layer with Levenberg-Marquart**

**Table 5. Implementation results of ICA algorithm**

| Dataset | Learning Method | No. of neurons in hidden layer | MSE | Accuracy |
|---|---|---|---|---|
| German | ICA-ANN | 25 | 0.141 | 79.10% |
| | LMBP-ANN | 21 | 0.189 | 77.10% |

The MSE changes to the number of neurons within the hidden layer with ICA, are shown in figure 7 for German dataset when minimum MSE has been occurred.

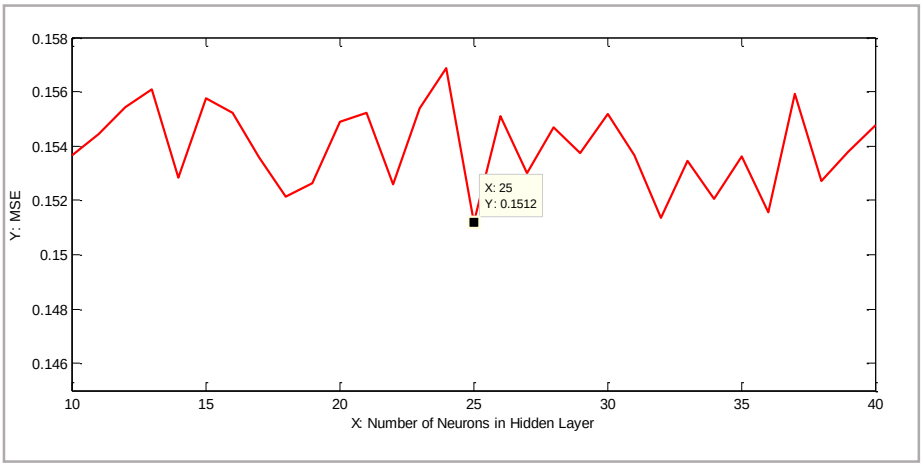


**Figure 7. Optimum number of neurons in the hidden layer with ICA**

The ICA-ANN models achieving the lowest MSE and best Accuracy are the ones used Tan-sigmoid and Log-sigmoid as activation functions in hidden and output layers respectively. Figure 8 shows the learning curve (error convergence) of ICA-ANN model. As mentioned before, the number of generations (max iteration) is considered as 200, but in order to overcome the problem of over-fitting

in neural network model, a validation check criterion with a value of 5 is determined. Therefore, learning algorithms are stopped upon achieving either of these criteria.

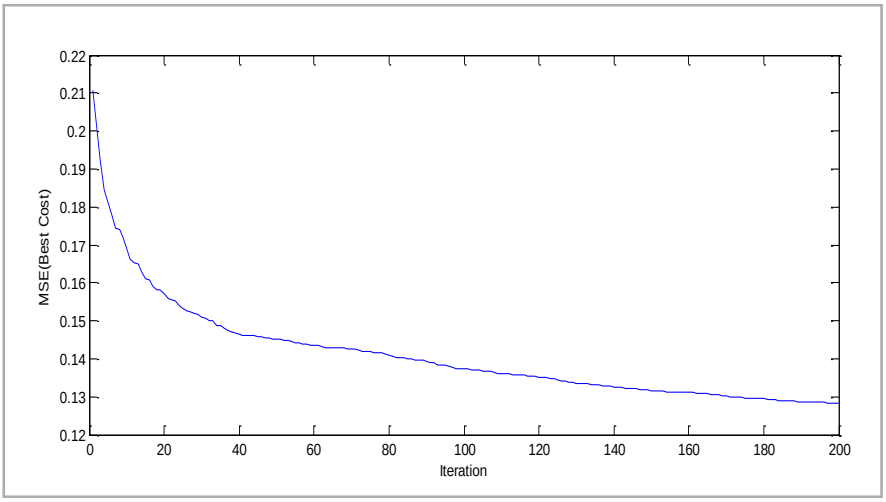


**Figure 8. Learning curve (error convergence) of ICA-ANN model**

From the results shown in Tables 5, when the performance of ICA-ANN and LMBP-ANN models is compared, ICA-ANN can perform better (the lowest MSE and best Accuracy), meaning that ICA is very effective in training ANNs.

## 8. Conclusion

The use of ANNs has been shown to be an effective technique. Among the different neural networks, MLP was used for evaluating learning algorithms. It is very important to choose a proper algorithm for training a neural network, so this paper presents a comparison of various BP learning algorithms and ICA with a special learning ratio and activation functions. BP algorithms update weights and biases in the direction of the negative gradient. ANNs training with BP algorithms is usually faced certain drawbacks such as very slow convergence and may trap in the local minimum. In contrast, weights and biases in ICA-ANN are represented by country position a possible answer in the d-dimensional space of the problem, whereas the size of the search space is the total number of weights and biases. Experimental results showed that the LM algorithm achieved better performance than all other BP algorithms in training MLP neural networks. Comparing these results with ICA as a training algorithm showed the superiority of ICA.

## References

[1] S. M. Aquil Burney, "*Tahseen A. Jilani, time series forecasting using artificial neural network methods for karachi stock exchange*", A project at department of computer science, university of Karachi, (2002).
[2] Y. Shang, B.W. Wah, "*Global Optimization for Neural Networks Training*", IEEE Computer. 29 (1996) 45-54.
[3] A. Abraham, "*Meta learning evolutionary artificial neural networks*", Neuro Computing. 56 (2004) 1-38.
[4] Jiancheng Fang, Zhongyu Wang, "*The combination of immune evolution and neural network for nonlinear time series forecasting Sixth International Symposium on Instrumentation and Control Technology: Sensors, Automatic Measurement, Control and Computer Simulation*", Beijing, China, (2006).
[5] Haibin Duan, Linzhi Huang, "*Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning*", Neurocomputing, (2014).

[6] A. Iftikhar, M.A. Ansari, S. Mohsin, "*Performance Comparison between Back propagation Algorithms Applied to Intrusion Detection in Computer Network Systems*", 9th WSEAS International Conference on Neural Networks. (2008) 231-236.
[7] R. Noori, M.S. Sabahi, A.R. Karbassi, "*Evaluation of PCA and Gamma test techniques on ANN operation for weekly solid waste predicting*", J. Environ. Manage. 91 (2010) 767-771.
[8] Mohammad Ali Ahmadi, Mohammad Ebadi, Amin Shokrollahi, Seyed Mohammad Javad Majidi. "*Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir*", Applied Soft Computin.13 (2013) 1085-1095.
[9] Shahram Mollaiy Berneti, Mehdi Shahbazian, "*an imperialist competitive algorithm-artificial neural network method to predict oil flow rate of the wells*", International Journal of Computer Applications. 26 (2011) 975-8887.
[10] S.M. Hosseini, A. Al Khaed. "*A survey on the Imperialist Competitive Algorithm metaheuristic*", Implementation in engineering domain and directions for future research. 24 (2014) 1078-1094.
[11] S. Mirjalili, A. Safa Sadiq, "*Magnetic optimization algorithm for training multi-layer perceptron*", in: IEEE International Conference on Industrial and Intelligent Information, Indonesia. 2 (2011) 42-46.
[12] M.C.P.de Souto, A. Yamazaki, T.B. Ludernir, "*Optimization of neural network weights and architecture for odor recognition using simulated annealing*", Proceedings of the 2002 International Joint Conference on Neural Networks. 1 (2002) 547–552.
[13] K. Hornick, M. Stinchcombe, H. White, "*Multilayer feed forward networks are universal approximators*", Neural Networks 2 (1989) 359–366.
[14] N. García-Pedrajas, C. Hervás-Martínez, J. Mu˜noz-Perez, "*A cooperative co-evolutionary model for evolving artificial neural networks*", IEEE Transactions on Neural Networks. 14 (2003) 575–596.
[15] T. Ganesan, I. Elamvazuthi, P. Vasant, "*Solving engineering optimization problems with the Karush–Kuhn–Tucker Hopfield neural networks*", International Review of Mechanical Engineering. 5 (2011) 1333-1339.
[16] S. Verma, B.D. Huey, D.J. Burgess, "*Scanning probe microscopy method for nanosuspension stabilizer selection*", Langmuir. 25 (2009) 12481-12487.
[17] Funahashi, K.I., "*On the Approximate Realization of Continuous Mappings by Neural Networks*", Neural Networks. 2 (1989) 183-192.
[18] Hornik, K., "*Approximation Capabilities of Multilayer Feedforward Networks*", Neural Networks. 4 (1991) 251-257.
[19] Cybenko, G., "*Approximation by Super positions of Sigmoidal Function*", Math. Contr. Signals Syst. 2 (1989) 303-314.
[20] K. Levenberg, "*A Method for the Solution of Certain Nonlinear Problems in Least Squares*", Quarterly of Applied Mathematics. 2 (1944) 164-168.
[21] M.S. Mirtalaei, M. Saberi, O.K. Hussain, B. Ashjari, F.K. Hussain, "*A trust-based bio-inspired approach for credit lending decisions*", Computing. 94 (2012) 541-577.
[22] Xinying Miao, Jinkui Chu, Linghan Zhang, Jing Qiao., "*An Evolutionary Neural Network Approach to Simple Prediction of Dam Deformation*", Journal of Information & Computational Science. 10 (2013) 1315–1324.
[23] Kunlei, Lian, Chaoyong, Zhang, Liang, Gaoa, "*Single row facility layout problem using an imperialist competitive algorithm*", Proceedings of the 41st International Conference on Computers & Industrial Engineering
[24] E. Atashpaz-Gargari, C. Lucas., "*Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition*", IEEE Congress on Evolutionary Computation. (2007) 4661-4667.